ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (29-36), Month: April-June 2014, Available at: www.researchpublish.com

Managing Trust Relationships in Peer 2 Peer Systems

R.S.SINJU
PG STUDENT, DEPARTMENT OF COMPUTER SCIENCE,
PONJESLY COLLEGE OF ENGINEERING
NAGERCOIL, TAMILNADU, INDIA

C.FELSY
ASST.PROF, DEPARTMENT OF COMPUTER SCIENCE,
PONJESLY COLLEGE OF ENGINEERING,
NAGERCOIL, TAMILNADU, INDIA

Abstract: **Peer-2-Peer system exposes them to malicious activity. Peer-2-Peer system means computer in the system can act as both client and server. In a Peer-2-Peer network, the peers are computer systems which are connected to each other via the internet. Files can be shared directly between systems on the network without the need of a central server. Building trust relationships among peers can decrease the attacks of malicious peers. A good peer uploads authentic files and gives fair recommendations. A malicious peer performs both service and recommendation-based attacks. Uploading a virus infected (or) an inauthentic file is a service based attack. Giving a misleading recommendation intentionally is a recommendation based attack. Self Organizing Trust Model (SORT) detects the service based attack and recommendation based attack. If one peer wants to upload/download file from another peer means peer will send the query to peer that interacted in the past for learn the trust information of other peers. So, neighbouring node will give the recommendation to peer. Based on the recommendation only Peer decides whether the node is good (or) malicious. Find the node is malicious node means peer will not interact with malicious node. Isolate the malicious node from the network. Find the node is good means peer interact with good peer. Peer stores a separate history of interactions for each Acquaintance.**

***Keywords*: Peer-to-peer systems, trust management, reputation, security.**

I. INTRODUCTION

Open nature of peer-to-peer systems exposes them to malicious activity. Building trust relationships among peers can mitigate attacks of malicious peers. This paper presents distributed algorithms that enable a peer to reason about trust worthiness of other peers based on past interactions and recommendations. Peers create their own trust network in their proximity by using local information available and do not try to learn global trust information. Two contexts of trust, service, and recommendation contexts are defined to measure trustworthiness in providing services and giving recommendations.

Interactions and recommendations are evaluated based on importance, recentness, and peer satisfaction parameters. Additionally, recommender's trustworthiness and confidence about a recommendation are considered while evaluating recommendations. Simulation experiments on a file sharing application show that the proposed model can mitigate attacks on 16 different malicious behavior models. In the experiments, good peers were able to form trust relationships in their proximity and isolate malicious peers. Peer to Peer (P2P) systems rely on collaboration of peers to accomplish tasks. Ease of performing malicious activity is a threat for security of P2P systems. Creating long-term trust relationships among peers can provide a more secure environment by reducing risk and uncertainty in future P2P interactions.

However, establishing trust in an unknown entity is difficult in such a malicious environment. Furthermore, trust is a social concept and hard to measure with numerical values. Metrics are needed to represent trust in computational models.

Classifying peers as either trustworthy or untrustworthy is not sufficient in most cases. Metrics should have precision so peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trustworthiness a challenge. In the presence of an authority, a central server is a preferred way to store and manage trust information..g., eBay. The central server securely stores trust information and defines trust metrics. Since there is no central server in most P2P systems, peers organize themselves to store and manage trust information about each other [2], [3]. Management of trust information is dependent to the structure of P2P network.

Managing trust is a problem of particular importance in peer-to-peer environments where one frequently encounters unknown agents. Existing methods for trust management that are based on reputation focus on the semantic proper- ties of the trust model. They do not scale as they either rely on a central database or require maintaining global knowledge at each agent to provide data on earlier interactions. In this paper we present an approach that addresses the problem of reputation-based trust management at both the data management and the semantic level. We employ at both levels scalable data structures and algorithms that require no central control and allow assessing trust by computing an agents reputation from its former interactions with other agents. There are no well defined methods for managing trust relationships in p2p systems. The DHT based approaches are only suited for structured p2p networks not for unstructured p2p networks. Some of the existing methods introduce central authority in p2p networks which may collapse p2p nature. Every agent must keep rather complex and very large data structures that represent a kind of global knowledge about the whole network.

This paper presents distributed algorithms that enable a peer to reason about trustworthiness of other peers based on past interactions and recommendations. Peers create their own trust network in their proximity by using local information available and do not try to learn global trust information.

Two contexts of trust, service, and recommendation contexts are defined to measure trustworthiness in providing services and giving recommendations. Self-Organizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers in their proximity.

In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers. An acquaintance is always preferred over a stranger if they are equally trustworthy. Using a service of a peer is an interaction, which is evaluated based on weight (importance) and recentness of the interaction, and satisfaction of the requester. An acquaintance's feedback about a peer, re commendation, is evaluated based on recommender's trust worthiness. It contains the recommender's own experience about the peer, information collected from the recommender's acquaintances, and the recommender's level of confidence in the recommendation. If the level of confidence is low, the recommendation has a low value in evaluation and affects less the trustworthiness of the recommender.

SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric.

One peer is marked as trusted by SORT and if it is turned OFF from network, there is a possibility to another malicious peer takes its position and act as trusted peer. This can be avoided by Auto Update Mechanism.

II. RELATED WORK AND CONTRIBUTION

The simulation runs as cycles. Each cycle represents a period of time. Downloading a file is an interaction. A peer sharing files is called an uploader. A peer downloading a file is called a downloader. The set of peers who downloaded a file from a peer are called downloaders of the peer. An ongoing download/ upload operation is called a session. Simulation parameters are generated based on results of several empirical studies [6], [7] to make observations realistic.

A file search request reaches up to 40 percent of the network and returns online uploaders only. A file is downloaded from one uploader to simplify integrity checking. All peers are assumed to have antivirus software so they can detect infected files Four different cases are studied to understand effects of trust calculation methods under attack conditions:

No trust. Trust information is not used for uploader selection. An uploader is selected according to its bandwidth. This method is the base case to understand if trust is helpful to mitigate attacks.

No reputation query. An uploader is selected based on trust information but peers do not request recommendations from other peers. Trust calculation is done based on SORT equations but reputation (r) value is always zero for a peer. This method will help us to assess if recommendations are helpful.

SORT. In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers.

Flood reputation query. SORT equations are used but a reputation query is flooded to the whole network. This method will help us to understand if getting more recommendations is helpful to mitigate attacks.

In SORT, to evaluate interactions and recommendations better, importance, recentness, and peer satisfaction parameters are considered. Recommender's trustworthiness and confidence about recommendation are considered when evaluating recommendations. Additionally, service and recommendation contexts are separated. This enabled us to measure trustworthiness in a wide variety of attack scenarios. Most trust models do not consider how interactions are rated and assume that a rating mechanism exists. In this study, we suggest an interaction rating mechanism on a file sharing application and consider many real-life parameters to make simulations more realistic.

A good peer uploads authentic files and gives fair recommendations. A malicious peer (attacker) performs both service and recommendation-based attacks. Four different attack behaviors are studied for malicious peers: naive, discriminatory, hypocritical, and oscillatory behaviours. A non-malicious network consists of only good peers. A malicious network contains both good and malicious peers.

The satisfaction parameter is calculated based on following variables: The ratio of average bandwidth (AveBw) and agreed bandwidth (AgrBw) is a measure of reliability of an uploader in terms of bandwidth. The ratio of online (OnP) and offline (OffP) periods represents availability of an uploader.

A. Network architecture

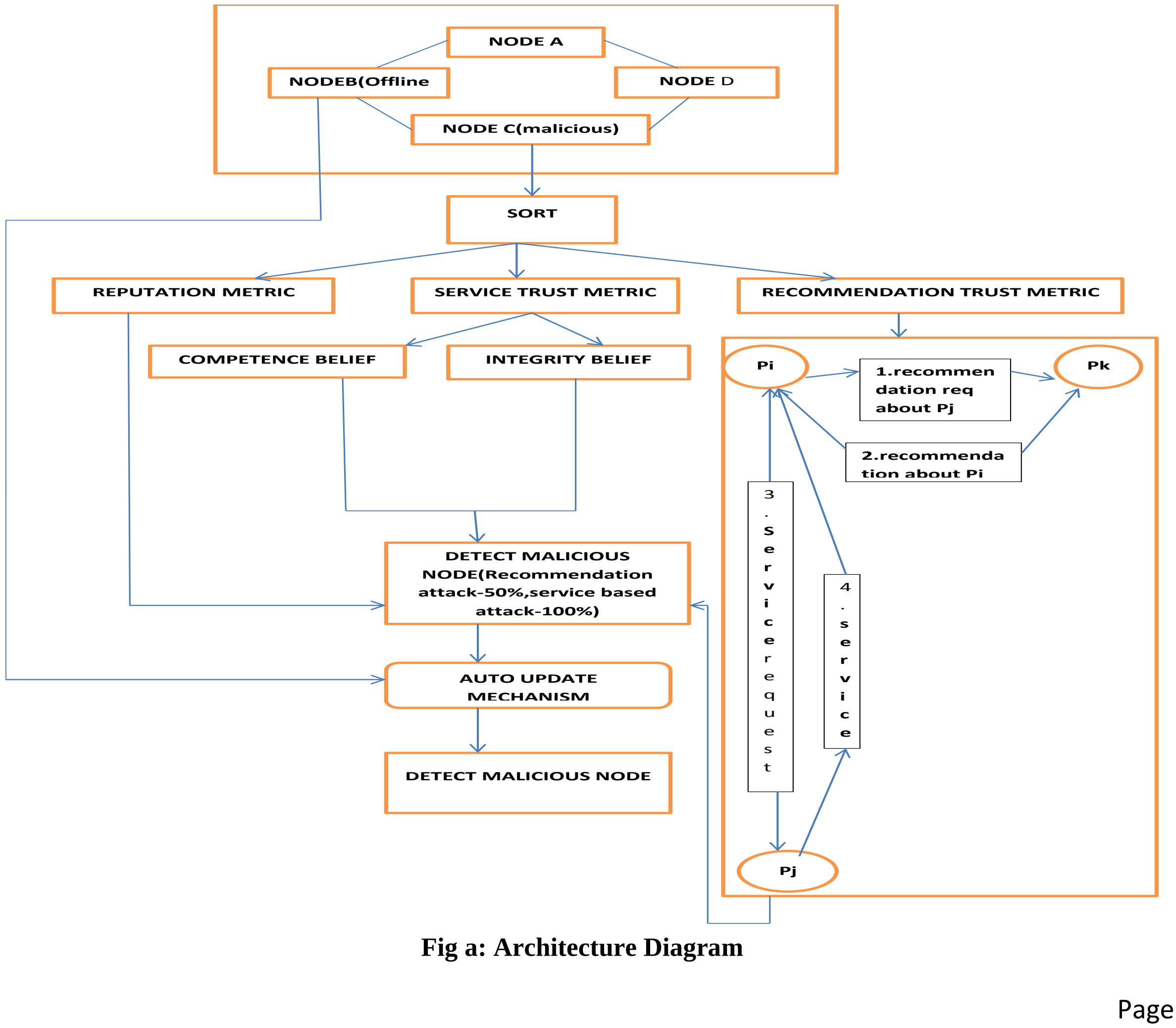


Fig a: Architecture Diagram

Downloading a file is an interaction. A peer sharing files is called an uploader. A peer downloading a file is called a downloader. The set of peers who downloaded a file from a peer are called downloaders of the peer. An ongoing download/ upload operation is called a session.

A good peer uploads authentic files and gives fair recommendations. A malicious peer (attacker) performs both service and recommendation-based attacks. Four different attack behaviours are studied for malicious peers: naive, discriminatory, hypocritical, and oscillatory behaviours. A non malicious network consists of only good peers. A malicious network contains both good and malicious peers.

SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases. Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively.

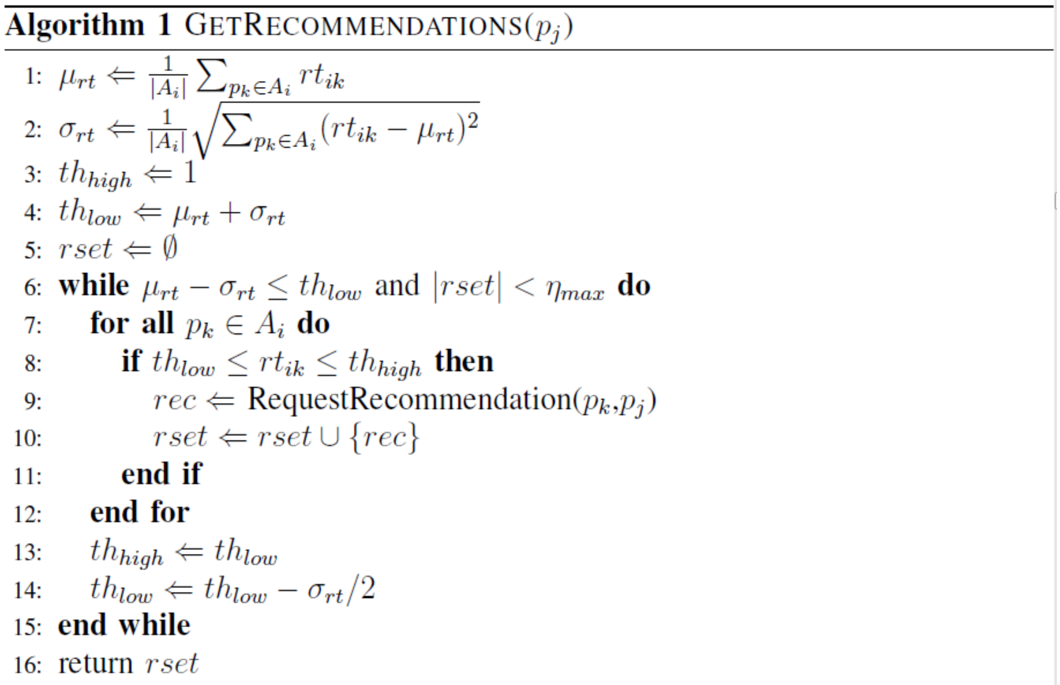
The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric.

Assume that pi wants to get a particular service. pj is a stranger to pi and a probable service provider. To learn pj's reputation, pi requests recommendations from its acquaintances. Assume that pk sends back a recommendation to pi. After collecting all recommendations, pi calculates rij.

Then, pi evaluates pk's recommendation, stores results in RHik, and updates rtik. Assuming pj is trustworthy enough, pi gets the service from pj. Then, pi evaluates this interaction and stores the results in SHij, and updates stij.

One peer is marked as trusted by SORT and if it is turned off from network,there is a possibility to another malicious peer takes its position and act as trusted peer.this can be avoided by the Auto update mechanism.

B. Algorithm Design and Implementation



Algorithm 1 GetRecommendations(p_j)

```
$\mu_{rt} \Leftarrow \frac{1}{|A_i|} \sum_{p_k \in A_i} rt_{ik}$
$\sigma_{rt} \Leftarrow \frac{1}{|A_i|} \sqrt{\sum_{p_k \in A_i} (rt_{ik} - \mu_{rt})^2}$
$th_{high} \Leftarrow 1$
$th_{low} \Leftarrow \mu_{rt} + \sigma_{rt}$
$rset \Leftarrow \emptyset$
while $\mu_{rt} - \sigma_{rt} \leq th_{low}$ and $|rset| < \eta_{max}$ do
    for all $p_k \in A_i$ do
        if $th_{low} \leq rt_{ik} \leq th_{high}$ then
            $rec \Leftarrow$ RequestRecommendation($p_k$,$p_j$)
            $rset \Leftarrow rset \cup \{rec\}$
        end if
    end for
    $th_{high} \Leftarrow th_{low}$
    $th_{low} \Leftarrow th_{low} - \sigma_{rt}/2$
end while
return $rset$
```

Topology creation is creating a network and maintaining communication among various nodes in peer to peer network which helps us to share the data. Create different nodes in proper name, ip address and port number for data communication. The node is added to give the name of the node, ip address and port address of that node. If the entire node adds successfully to display the node connection frames.

Creating long-term trust relationships among peers can provide a more secure environment by reducing risk and uncertainty in future P2P interactions. However, establishing trust in an unknown entity is difficult in such a malicious environment. Furthermore, trust is a social concept and hard to measure with numerical values. Metrics are needed to represent trust in computational models. Classifying peers as either trustworthy or untrustworthy is not sufficient in most cases.

Metrics should have precision so peers can be ranked according to trustworthiness. Interactions and feedbacks of peers provide information to measure trust among peers. Interactions with a peer provide certain information about the peer but feedbacks might contain deceptive information. This makes assessment of trustworthiness a challenge.

Self-Organizing Trust model (SORT) that aims to decrease malicious activity in a P2P system by establishing trust relations among peers. Each peer develops its own local view of trust about the peers interacted in the past. In this way, good peers form dynamic trust groups in their proximity and can isolate malicious peers. In SORT, peers are assumed to be strangers to each other at the beginning. A peer becomes an acquaintance of another peer after providing a service, e.g., uploading a file. If a peer has no acquaintance, it chooses to trust strangers.

SORT defines three trust metrics. Reputation metric is calculated based on recommendations. It is important when deciding about strangers and new acquaintances. Reputation loses its importance as experience with an acquaintance increases.

Service trust and recommendation trust are primary metrics to measure trustworthiness in the service and recommendation contexts, respectively. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations. When calculating the reputation metric, recommendations are evaluated based on the recommendation trust metric.

Creating trust relationship is based upon two contexts of trust. They are Service Context, Recommendation Context. The service trust metric is used when selecting service providers. The recommendation trust metric is important when requesting recommendations.

When pi searches for a particular service, it gets list of service providers. Considering a file sharing application, pimay download a file from either one or multiple uploaders. With multiple uploaders, checking integrity is a problem since any file part downloaded from an uploader might be inauthentic..

Assume that pi wants to get a particular service. pj is a stranger to pi and a probable service provider. To learn pj's reputation, pi requests recommendations from its acquaintances. Assume that pk sends back a recommendation to pi. After collecting all recommendations, pi calculates rij.

Then, pi evaluates pk's recommendation, stores results in RHik, and updates rtik. Assuming pj is trustworthy enough, pi gets the service from pj. Then, pi evaluates this interaction and stores the results in SHij, and updates stij.

III. PERFORMANCE ANALYSIS AND RESULTS

A. Performance Analysis

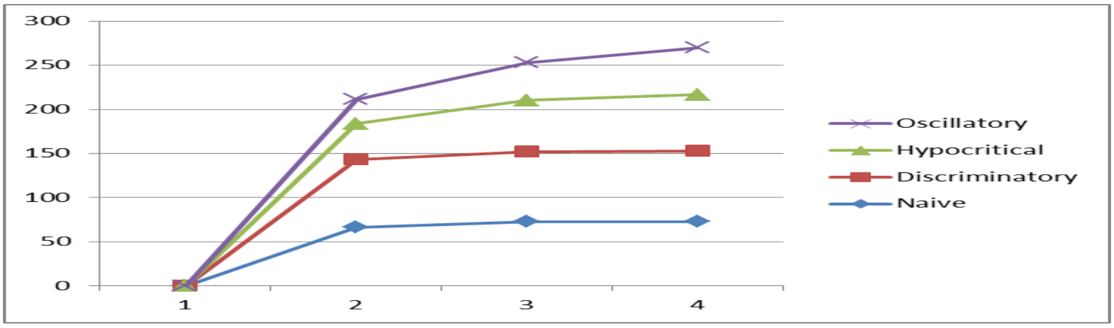


Fig b: Existing System(10% Malicious)

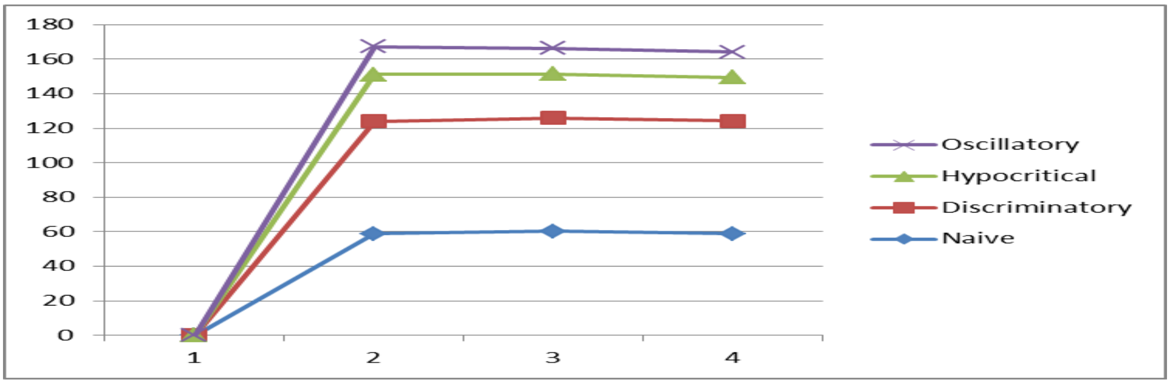


Fig c: Proposed System (50% Malicious)

Analysis on Individual Attackers

This section explains the results of experiments on individual attackers. For each type of individual attacker, two separate network topologies are created: one with 10 percent malicious and one with 50 percent malicious. Each network topology is tested with four trust calculation methods. In the experiments, a hypocritical attacker behaves malicious in 20 percent of all interactions. A discriminatory attacker selects 10 percent of all peers as victims. An oscillatory attacker behaves good for 1,000 cycles and malicious for 100 cycles.

Service-based attacks

Table shows the percentage of service-based attacks prevented by each trust calculation method. When a malicious peer uploads an infected/ inauthentic file, it is recorded as a service-based attack. Number of attacks in No Trust method is considered as the base case to understand how many attacks can happen without using trust information. Then, number of attacks observed for each trust calculation method is compared with the base case to determine the percentage of attacks prevented. In the table, NoRQ and FloodRQ denote “No reputation query” and “Flood reputation query” methods, respectively.

Recommendation-based attacks

In the simulations, when a malicious peer gives a misleading recommendation, it is recorded as a recommendation-based attack. Fig. 2 shows the rate of recommendation-based attacks in the 10 percent malicious network. When SORT is used, peers form their own trust network with time and do not request recommendations from untrustworthy peers. Therefore, SORT can effectively mitigate recommendation-based attacks with time. In FloodRQ method, peers collect more recommendations from both acquaintances and strangers.

Distribution of trust metrics

Peers with higher capabilities (network bandwidth, online period, and number of shared files) can finish more interactions successfully. Thus, they generally have better reputation and service trust values. Recommendation trust values are not directly related to peer capabilities since giving a recommendation does not require high capabilities.

Analysis on Individual Pseudospoofers

This section explains the results of experiments on individual pseudospoofers. Pseudospoofers change their pseudonyms after every 1,000 cycles.

Service-based attacks

Table shows the attack prevention ratio for individual pseudospoofers. The values obtained by comparing the base case with each trust calculation method. After every pseudonym change, attackers become strangers to others. This behavior has two effects: 1) Pseudospoofers clear their bad history. Hence a good peer may interact with them when it cannot find more reliable uploaders, which increases attacks. 2) Pseudospoofers become more isolated from good peers. They lose their ability to attract good peers with time, which decreases attacks.

Recommendation-based attacks

Comparing to nonpseudonym changing individual attackers, recommendation-based attacks decrease due to the second effect. Attackers become more isolated from good peers after every pseudonym change and get less recommendation requests. Therefore, their recommendation-based attacks sharply decrease after every pseudonym change in a 10 percent malicious network. This situation is slightly different in a 50 percent malicious network. The good peers need to interact with more strangers since they can hardly find each other. Hence attackers can distribute more misleading recommendations.

Distribution of trust metrics

Since pseudospoofers clear their old history in every 1,000 cycles, they cannot gain a high reputation among good peers. This situation is same for service trust and recommendation trust metrics. Average reputation, service trust, and recommendation trust values of pseudospoofers remain under 0.01 value in most simulations.

Analysis on Collaborators

Collaboration of attackers generally makes attack prevention harder. This section presents experimental results on collaborators. Collaborators form teams of size 50 and launch attacks as teams. We first tried teams of size 10 but this was not enough to benefit from collaboration and results were close to individual attackers. Hence team size is increased to observe effects of collaboration better.

Service-based attacks

Table shows the percentage of attacks prevented by each method. Attacks of naïve collaborators can be prevented by 60 percent or more. Naive collaborators are identified by good peers after the first interaction so they are not asked for recommendations.

Recommendation Based Attacks

In a 50 percent malicious network, attacks of hypocritical and oscillatory collaborators can be contained on a level but cannot be decreased to an acceptable level. They can continue to disseminate misleading recommendations due to their large number. Discriminatory collaborators can disseminate more misleading recommendations than others since they are trusted by 90 percent of all peers. Discriminatory collaborators constitute 50 percent of all peer population while victims are 10 percent of all population.

Trust Metric

Good peers can maintain higher reputation than hypocritical collaborators in the 10 percent malicious network. In 50 percent malicious network setup, collaborators gain higher reputation values and decrease reputation of good peer.

Analysis on Collaborating Pdesudospoofers

This section presents the results of experiments on collaborating pseudospoofers. Collaborating pseudospoofers are assumed to change their pseudonyms in every 1,000 cycles using an external synchronization method.

Service-based attacks

Table shows the percentage of attacks prevented by each trust calculation method. The results verify our observations. In naive and discriminatory behaviors, changing pseudonym causes the first effect: collaborators clear bad history and get more attack opportunities. Therefore, attack prevention ratio drops for these collaborators.

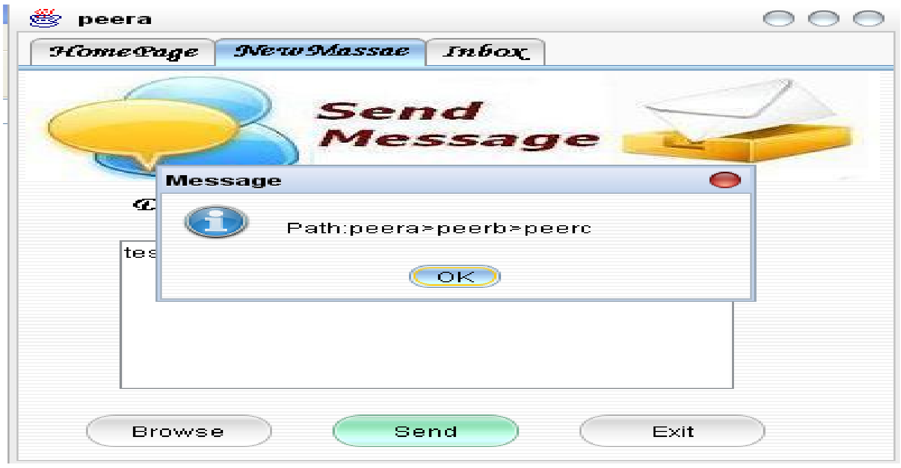
Recommendation-based attacks

As in individual pseudospoofers, collaborating pseudospoofers are isolated more from good peers after every pseudonym change. They get less recommendation requests and thus they can do nearly zero recommendation-based attacks in 10 percent malicious network.

Trust metrics

Like individual pseudospoofers, collaborating pseudospoofers cannot gain high reputation, service trust, or recommendation trust values since they lose reputation after every pseudonym change. Due to their low recommendation trust values, collaborators are not asked for recommendations by good peers. Therefore, they can distribute small number of misleading recommendations.

B. Results



Path:peera>peerb>peerc

Fig d: Message Communication

The data can be communicated effectively without any malicious activity.The Self Organizing Trust Model finds the malicious node by itself with service and recommendation context.So that the communication is made more effective without the attack of malicious node in the network.Setting B as 'malicious Node'.Send data from 'a' to 'c' .Here the message is only Received to 'b'. Since 'B' is malicious.This path is added as invalid. Now I am going to send message from 'D' to B .Here the message received to 'B' Since the intermediate node 'C' is not malicious. The path is added as valid.

IV. CONCLUSION

SORT mitigated both service and recommendation-based attacks in most experiments. However, in extremely malicious environments such as a 50 percent malicious network, collaborators can continue to disseminate large amount of misleading recommendations. Another issue about SORT is maintaining trust all over the network. These issues might be studied as a future work to extend the trust model. Using trust information does not solve all security problems in P2P systems but can enhance security and effectiveness of systems.

REFERENCES

[1] AhmetBurakCan and Bharat(2013),'A Self-Organizing Trust Model for Peer-to-Peer Systems'IEEE Trans.Dependable and Secure Computing,vol 10,No.1.

[2] Aberer.K and Despotovic.Z(2001), 'Managing Trust in a Peer-2-Peer Information System' Proc. 10th Intl Conf. Information and Knowledge Management (CIKM).

[3] Kamvar.S, Schlosser.M, and Garcia-Molina.H,(2003) 'The (Eigentrust) Algorithm for Reputation Management in P2P Networks' Proc. 12th World Wide Web Conf. (WWW).

[4] SelcukA.A ,Uzun.E, and Pariente.M.R(2004), 'A Reputation-Based Trust Management System for P2P Networks' Proc. IEEE/ACM Fourth Int'l Symp. Cluster Computing and the Grid (CCGRID).

[5] Zhou. R, Hwang. K, and Cai. M(2008), 'Gossiptrust for Fast Reputation Aggregation in Peer-to-Peer Networks' IEEE Trans. Knowledge and Data Eng., vol. 20, no. 9.

[6] Abdul-Rahman. A and Hailes.S(2008), 'Supporting Trust in Virtual Communities' Proc. 33rd Hawaii Int'l Conf. System Sciences (HICSS).

[7] Yu. B and Singh.M(2000), 'A Social Mechanism of Reputation Management in Electronic Communities' Proc. Cooperative Information Agents (CIA).